BAB 2
TINJAUAN PUSTAKA

2.1 Pofil Instansi

2.1.1 Sejarah Kartin Collection

Konveksi tas Kartin Collection didirikan oleh seorang wirausahawan yang bernama bapak Agus Supianto. Konveksi tersebut berdiri tahun 1990 di Jln. Karang Tineung No.1 Bandung . Tujuan pendirian usaha pembuatan tas dari bahan dolbi maupun bahan lain ini dilihat dari potensi masyarakat akan bersifat konsumtif, bapak Agus akhirnya mendirikan suatu konveksi pembuatan tas. Tidak hanya dilihat dari masyarakatnya saja pendirian usaha tersebut difungsikan untuk memenuhi kebutuhan keluarga bapak Agus. Akhirnya sampai saat ini konveksi tersebut dapat berkembang pesat dengan kiat-kiat seorang wirausaha yang dimiliki oleh bapak Agus yaitu percaya diri, optimis, dan tekad yang besar untuk menggapai suatu impian dalam berwirausaha. Ditambah etos kerja dan ulet dalam mengerjakan usaha yang digelutinya.

2.1.2 Logo

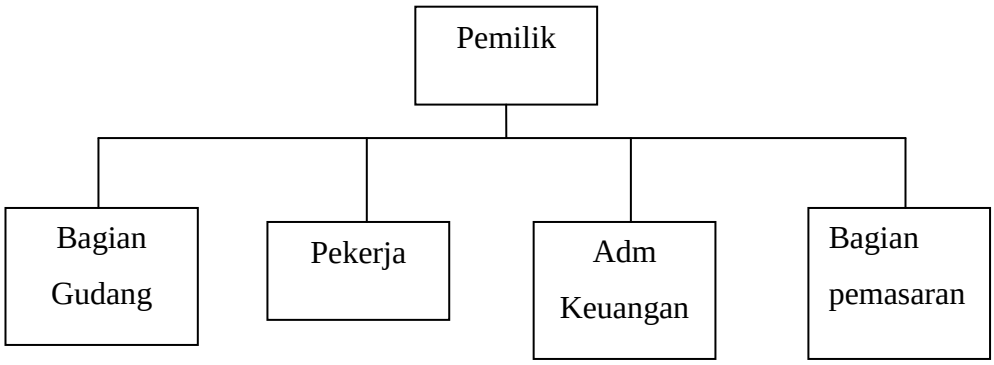
Berikut ini adalah logo dari konveksi tas Kartin Collection yang dapat dilihat pada gambar 2.1 dibawah ini :



Gambar 2. 1 Logo Kartin Collection

2.1.3 Struktur Organisasi

Berikut ini adalah struktur organisasi konveksi tas kartin Collection yang dapat dilihat pada gambar 2.2 dibawah ini :



Gambar 2. 2 Struktur Organisasi kartin Collection

Job description :

1. Pemilik bertanggung jawab penuh terhadap Kartin Collection, tugas dan tanggung jawabnya sebagai berikut:
   a. Membuat perencanaan, strategi dan kebijakan yang menyangkut operasi Kartin Collection.
   b. Menyusun anggaran kebutuhan persediaan bahan baku.
   c. Melakukan kontrol secara keseluruhan di Kartin Collection.
   d. Memegang kendali atas keputusan penting yang bersifat umum berkaitan dengan keuangan.
2. Bagian Gudang memiliki tugas dan tanggung jawab sebagai berikut :
   a. Mengecek ketersedian bahan baku.
   b. Membeli persedian bahan baku.
   c. Memberikan bahan baku kepada pekerja.
3. Pekerja memeliki tugas dan tanggung jawab sebagai berikut:
   a. Membuat tas sesuai perintah.
   b. Menyelesaikan pembuatan tas tepat waktu.

4. Bagian Adm keuangan bertugas terhadap transaksi pelanggan, tugas dan tanggung jawab sebagai berikut :
   a. Bertanggung jawab terhadap hal-hal yang menyangkut keuangan.
   b. Menghitung pemasukan dan pengeluaran setiap bulannya.
5. Bagian pemasaran memeliki tugas dan tanggung jawab untuk memasarkan dan memdistribusikan produk tas kepada pelanggan.

2.1.4 Visi dan Misi

Visi dari Kartin Collection ini adalah menjadikan perusahaan konveksi yang terbaik, dengan pengerjaan pesanan yang tepat waktu dan mampu melayani permintaan pesanan sesuai dengan apa yang diinginkan oleh setiap konsumen. baik dari segi pelayanan, kualitas, maupun kuantitas yang memuaskan, sedangkan misinya adalah Mengutamakan pelayanan pada kepuasan yang optimal bagi para pelanggan. Berperan aktif untuk meningkatkan kualitas dan produktivitas yang dapat memberikan kepuasan para pelanggan, karyawan dan mitra bisnis. mengembangkan sumber daya untuk dapat menghasilkan produk yang berkualitas dan memiliki mutu yang konsisten.

2.2 Landasan teori

Landasan teori yang berkaitan dengan materi atau teori yang digunakan sebagai acuan melakukan penenlitian. Landasan teori yang diuraikan merupakan hasil *studi literatur*, buku-buku, maupun situs internet.

2.2.1 Data

Data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya [5]. *Data* dapat digunakan sebagai *input* dan menghasilkan sebuah informasi. *Data* juga merupakan sesuatu yang belum memiliki arti dan masih membutuhkan suatu pengolahan. Dalam *data* terdapat himpunan *data* yang merupakan kumpulan dari objek dan atributnya. Atribut merupakan sifat atau karakteristik dari suatu objek yang biasanya dikenal

sebagai *variabel, field,* karakteristik atau fitur. Salah satu himpunan *data* adalah *record data,* yaitu *data* yang terdiri dari sekumpulan *record*, yang masing-masing terdiri dari satu set atribut yang tetap. Salah satu yang termasuk dalam tipe *data record* yaitu *data* transaksi. *Data* transaksi merupakan sebuah tipe khusus dari *record data*, dimana tiap *record* (transaksi) meliputi satu *set item* [6].

2.2.2 Basis Data

Basis data adalah mekanisme yang digunakan untuk menyimpan informasi atau *data*. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basis data, pengguna dapat menyimpan *data* secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basis data menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. *Data* pun harus mudah ditambahkan ke dalam basis data, dimodifikasi, dan dihapus [7].

Menurut Fathansyah [8], basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan *data* yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file*/ tabel/ arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

2.2.2.1 *Data* pada Basis Data dan Hubungannya

Ada 3 jenis data pada sistem *database,* yaitu [9] :

1. Data operasional dari suatu organisasi, berupa *data* yang tersimpan dalam basis data
2. Data masukan (*input data*), data dari luar sistem yang dimasukan melalui peralatan input (*keyboard*), yang dapat merubah data operasional.

3. Data kelauran (*output data*), berupa laporan melalui peralatan *output* sebagai hasil dari dalam sistem yang mengakses data operasional.

2.2.2.2 Keuntungan dan Kerugian Pemakaian Sistem *Database*

Keuntungan [9] :

1. Terpeliharanya keselarasan *data*
2. *Data* dapat dipakai secara bersama-sama.
3. Memudahkan penerapan standarisaasi dan batas-batas pengamanan.
4. Terpeliharanya keseimbangan atas perbedaan kebutuhan data dari setiap aplikasi.
5. Program /data *independent*.

Kerugian :

1. Mahal dalam implementasinya.
2. Rumit
3. Penanganan proses *recovery backup* sulit.
4. Kerusakan pada sistem basis data dapat mempengaruhi.

2.2.3 *Database Mangement System*

Kumpulan atau gabungan *database* dengan perangkat lunak aplikasi yang berbasis *database* tersebut dinamakan *Database Management System (DBMS). DBMS* merupakan koleksi terpadu dari *database* dan program–program komputer (utilitas) yang digunakan untuk mengakses dan memelihara *database*. Program-program tersebut menyediakan berbagai fasilitas operasi untuk memasukan, melacak, dan memodifikasi data kedalam *database*, mendefinisikan *data* baru, serta mengolah data menjadi informasi yang dibutuhkan (*DBMS* = *Database* + Program Utilitas) [8].

2.2.4 *Data Mining*

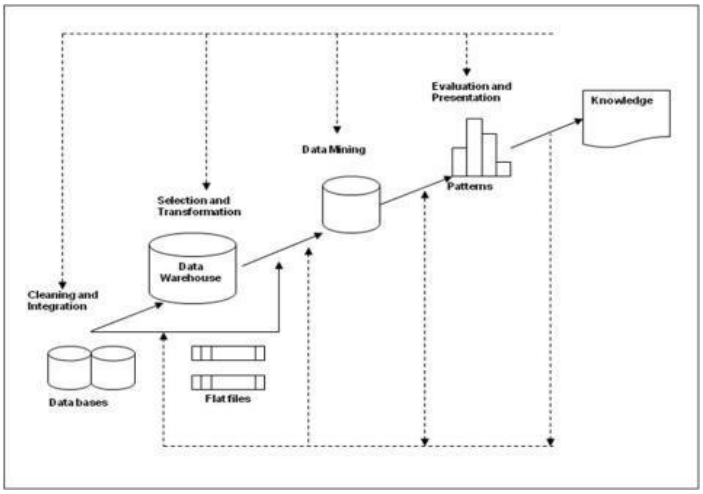
Data mining, sering juga disebut *knowledge discovery in database (KDD)*, adalah kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola atau hubungan dalam set *data* berukuran besar. Keluaran dari *data mining* ini bisa dipakai untuk memperbaiki pengambilan keputusan di masa depan. Sehingga istilah *pattern recognition* sudah tidak digunakan lagi [10].

Secara umum, definisi *data mining* dapat diartikan sebagai berikut [11]:

1. Proses penemuan pola yang menarik dari data yang tersimpan dalam jumlah besar.
2. Ekstrasi dari suatu informasi yang berguna atau menarik (*non-trivial*, implisit, sebelumnya belum diketahui potensi kegunaannya) pola atau pengetahuan dari data yang di simpan dalam jumlah besar.
3. Eksplorasi dari analisa secara otomatis atau semiotomatis terhadap data data dalam jumlah besar untuk mencari pola dan aturan yang berarti.

2.2.4.1 Konsep *Data Mining*

Data mining sangat diperlukan terutama dalam mengelola *data* yang sangat besar untuk memudahkan aktifitas *recording* suatu transaksi dan untuk proses *data warehousing* agar dapat memberikan informasi yang akurat bagi bagi pengguna *data mining*. Alasan utama *data mining* sangat dibutuhkan dalam industri informasi karena tersedianya data dalam jumlah yang besar dan semakin besarnya kebutuhan untuk mengubah data tersebut menjadi informasi dan pengetahuan yang berguna karena sesuai fokus bidang ilmu ini yaitu melakukan kegiatan mengekstraksi atau menambang pengetahuan dari data yang berukuran atau berjumlah besar. Informasi iniliah yang nantinya sangat berguna untuk pengembangan. Berikut adalah langkah-langkah dalam data mining [10] :



Gambar 2. 3 Konsep *Data Mining*

1. *Data cleaning* yaitu untuk menghilangkan *noise* data yang tidak konsisten.
2. *Data integration* yaitu menggabungkan beberapa *file* atau *database*.
3. *Data selection* yaitu data yang relevan dengan tugas analisis dikembalikan ke dalam *database* untuk proses *data mining*.
4. *Data transformation* yaitu data berubah atau bersatu menjadi bentuk yang tepat untuk menambang dengan ringkasan performa atau operasi agresi.
5. *Data mining* yaitu proses esensial dimana metode yang intelejen digunakan untuk mengekstrak pola *data*.
6. *Knowledge disccovery* yaitu proses esential dimana metode yang intelejen digunakan untuk mengekstrak pola *data*.
7. *Pattern evolution* yaitu untuk mengidentifikasi pola yang benar-benar menarik yang mewakili pengetahuan berdasarkan atas beberapa tindakan yang menarik.
8. *Knowledge presentation* yaitu gambaran teknik visualisasi dan pengetahuan digunakan untuk memberikan pengetahuan yang telah ditambah kepada *user*.

2.2.4.2 Metode *Data Mining*

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan, yaitu [12] :

1. Deskripsi

Terkadang peneliti dan analis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat mengumpulkan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden. deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelesan untuk suatu pola atau kecenderungan.

2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali *variabel* target estimasi lebih kearah numerik daripada ke arah kategori. Model dibangun dengan *record* lengkap yang menyediakan nilai dari *variabel* target sebagai nilai prediksi. Sebagai contoh, akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk kasus baru lainnya.

3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada dimasa mendatang. Contoh prediksi dalam bisnis dan penelitian adalah :

1. Prediksi harga beras dalam tiga bulan yang akan datang.
2. Prediksi persentase kenaikan kecelakaan lalu lintas tahun depan jika batas bawah dinaikan.

4. Klasifikasi

Dalam klasifikasi, terdapat terget variabel kategori. sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori yaitu: pendapatan tinggi, pendapatan sedang, dan pendapatan rendah. Contoh lain klasifikasi dalam bisnis dan penelitian adalah :

1. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau bukan.
2. Mendiagnosis penyakit seorang pasien untuk mendapatkan termasuk kategori penyakit apa.

5. Pengklusteran

Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. *Kluster* adalah kumpulan *record* yang memiliki kemiripan satu

dengan yang lainnya dan tidak memiliki kemiripan dengan *record-record* dalam kluster lain. Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya *variabel* target dalam pengklusteran. pengklusteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari *variabel target*. Akan tetapi, *algoritma* pengklusteran mencoba untuk melakukan pembagian terhadap keselurahan data menjadi kelompok-kelompok yang memiliki kemiripan (*homogen*), yang mana kemiripan *record* dalam suatu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal. Contoh pengklusteran dalam bisnis dan penelitian adalah :

1. Mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari suatu produk sebuah perusahaan yang tidak memiliki dana pemasaran yang besar.
2. Untuk tujuan audit akuntansi, yaitu melakukan pemisahan terhadap perilaku finansial dalam baik maupun mencurigakan.
3. Asosiasi

   Tugas asosiasi dalam *data mining* adalah menemukan atribut yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang pasar. Contoh asosiasi dalam bisnis dan penelitian adalah :

   1. Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respon positif terhadap penawaran *upgrade* layanan yang diberikan.
   2. Menemukan barang dalam supermarket yang dibeli secara bersamaan dan barang yang tidak pernah dibeli secara bersamaan.

2.2.5 *Association Rule*

Aturan *asosiasi (Association rule)* adalah salah satu teknik tentang ‘apa bersama apa’. Ini bisa berupa transaksi di supermarket, misalnya seseorang yang membeli susu bayi juga membeli sabun mandi. Di sini berarti susu bayi bersama sabun mandi. Karena awalnya berasal dari studi tentang *database* transaksi pelanggan untuk menentukan kebiasaan suatu produk dibeli bersama produk apa, maka aturan asosiasi juga sering dinamakan *Market Basket* [13].

Untuk mencari *association rule* dari suatu kumpulan data, tahap pertama yang harus dilakukan adalah mencari *frequent itemset* terlebih dahulu. *Frequent itemset* adalah sekumpulan *item* yang sering muncul secara bersamaan. Setelah semua pola *frequent itemset* ditemukan, barulah mencari aturan asosiatif atau aturan keterkaitan yang memenuhui syarat yang telah ditentukan.

Jika diasumsikan bahwa barang yang dijual di swalayan adalah semesta, maka setiap barang akan memiliki *Boolean variable* yang akan menunjukan keberadaannya atau tidak barang tersebut dalam satu transaksi atau satu keranjang belanja. Pola *Boolean* yang didapat digunakan untuk menganalisa barang yang dibeli secara bersamaan. Pola tersebut dirumuskan dalam sebuah *association rule.* Sebagai contoh konsumen biasanya akan membeli kopi dan susu yang ditujukan sebagai berikut :

Kopi → susu [*support* = 2%, *confidence* = 60%]

Nilai *support* 2% menunjukan bahwa keseluruhan dari total transaksi konsumen membeli kopi dan susu secara bersamaan yaitu sebanyak 2%. Sedangkan *confidence* 60% yaitu menunjukan bila konsumen membeli kopi dan pasti membeli susu sebesar 60%.

Penting tidaknya suatu aturan asosiatif dapat diketahui dengan dua parameter, yaitu *support* dan *confidence. Support* (nilai penunjang) adalah presentase kombinasi *item* tersebut dalam *database*, sedangkan *confidence* (nilai kepastian) adalah kuatnya hubungan antar-*item* dalam aturan asosiasi.

Dalam menentukan suatu *association rule*, terdapat suatu ukuran kepercayaan yang didapat dari hasil pengelolahan data dengan perhitungan tertentu. Umumnya ada dua ukuran, yaitu:

a. *Support* : suatu ukuran yang menunjukan seberapa besar tingkat dominasi suatu *itemset* dari keseluruhan transaksi. Ukuran ini menentukan apakah suatu *itemset* layak untuk dicari *confidence*-nya (misal, dari keseluruhan transaksi yang ada, seberapa besar tingkat dominasi suatu *item* yang menunjukan bahwa *item* A dan *item* B dibeli bersamaan).
b. *Confidence* : suatu ukuran yang menunjukan hubungan antara 2 *item* secara *conditional* (misal, menghitung kemungkinan seberapa sering *item* B dibeli oleh pelanggan jika pelanggan tersebut membeli sebuah *item* A).

Kedua ukuran ini nantinya berguna dalam menentukan kekuatan suatu pola dengan membandingkan pola tersebut dengan nilai minimum kedua parameter tersebut yang ditentukan oleh pengguna. Analisis asosiasi didefinisikan suatu proses untuk menemukan semua aturan asosiasi yang memenuhi syarat minimum untuk *support* (*minimum support*) dan syarat minimum untuk *confidence* (*minimum confidence*).

Nilai *support* sebuah *item* diperoleh dengan rumus sebagai berikut[6] :

$$Support(\mathrm{A}) = \frac{Jumlah\ transaksi\ mengandung\ A}{Total\ Transaksi} \mathrm{x}100\% \ldots\ldots \text{ Persamaan (2.1)}$$

Sementara itu, nilai *support* dari 2 *item* diperoleh dari rumus berikut:

$$Support\ (\mathrm{A}, \mathrm{B}) = \frac{\text{Jumlah transaksi yang mengandung A dan B}}{\text{Jumlah transaksi}} \mathrm{x}100\% \text{..Persamaan (2.2)}$$

Sedangkan nilai *confidence* dapat dicari setelah pola frekuensi munculnya sebuah *item* ditemukan. Rumus untuk menghitung *confidence* adalah sebagai berikut:

$$Confidence = \frac{\text{Jumlah transaksi yang mengandung A dan B}}{\text{Jumlah transaksi yang mengandung A}} \mathrm{x}100\% \ldots\ldots \text{Persamaan(2.3)}$$

2.2.6 Algoritma *FP-Growth*

Algoritma yang sama dengan Apriori, *FP-Growth* mulai dengan menghitung item tunggal sesuai dengan jumlah kemunculan item yang ada didalam *dataset*. Setelah proses penghitungan selesai maka akan dibuat struktur pohon pada tahap kedua. Pohon yang dibuat mulanya kosong yang nanti akan diisi dengan hasil dari dataset yang telah didapat sebelumnya. Kunci untuk mendapatkan struktur pohon yang bisa didapatkan dengan proses lebih cepat untuk mencari item set yang besar menjadi sedikit dengan di urutkan secara

descending dari frekuensi yang ada *dataset* tersebut. Masing-masing item yang tidak mencapai kebutuhan minimum dari *threshold* tidak dimasukkan kedalam pohon, tapi dikeluarkan secara efektif dari *dataset*.

FP-Tree merupakan struktur penyimpanan data yang dimampatkan. *FP-Tree* dibangun dengan memetakan setiap data transaksi ke dalam setiap lintasan tertentu dalam *FP-Tree.* Karena dalam setiap transaksi yang dipetakan, mungkin ada transaksi yang memiliki item yang sama, maka lintasannya memungkinkan untuk saling menimpa. Semakin banyak data transaksi yang memiliki item yang sama, maka proses pemampatan dengan struktur data *FP-Tree* semakin efektif. Kelebihan dari *FP-Tree* adalah hanya memerlukan dua kali pemindaian data transaksi yang terbukti sangat efisien. *FP-Growth* dapat dibagi menjadi 3 tahapan utama yaitu sebagai :

1. Tahap Pembangkitan *conditional pattern base.*

   Conditional Pattern Base merupakan sub *database* yang berisi *prefix path* dan *suffix pattern*. Pembangkitan *conditinal pattern base* didapatkan melalui *FP-Tree* yang telah dibangun sebelumnya.
2. Tahap Pembangkitan *Conditional FP-Tree*

   Pada tahap ini, *support count* dari setiap item pada setiap *conditional pattern base* dijumlahkan, lalu setiap item yang memiliki jumlah *support count* lebih besar atau sama dengan *minimum support count* ξ akan dibangkitkan dengan *conditional FP-Tree*.
3. Tahap Pencarian *frequent pattern*

   Apabila *conditional FP-Tree* merupakan lintasan tunggal (*single path*), maka didapatkan *frequent itemset* dnegan melakukan kombinasi item untuk setiap *conditional FP-Tree*. Jika bukan lintasan tunggal, maka dilakukan pembangkitan *FP-Growth* secara rekursif.

Berikut ini merupakan contoh kasus penerapan metode *association rule* menggunakan Algoritma *FP-Growth* :

Misalkan diberikan tabel data transaksi sebagai berikut dengan *minimum support count* = 2

Tabel 2.1 Data Transaksi

TID	*Items*
1	B
2	B, D
3	A, C, D, E
4	A, D, E
5	B, C
6	A, C
7	A

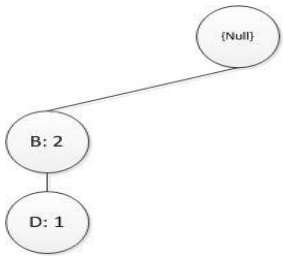
Yang mana nilai *support count* didapat dari : $\frac{15}{7}$ x 100% = 2.14 ≈ 2

Frekuensi kemunculan tiap item dapat dilihat pada tabel berikut:

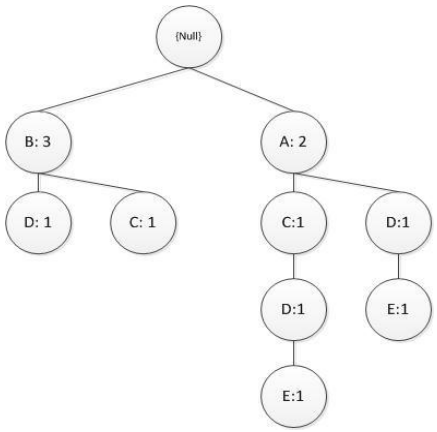
Tabel 2. 2 Frekuensi Kemunculan Tiap Karakter

Item	Frekuensi
A	4
B	3
C	3
D	3
E	2

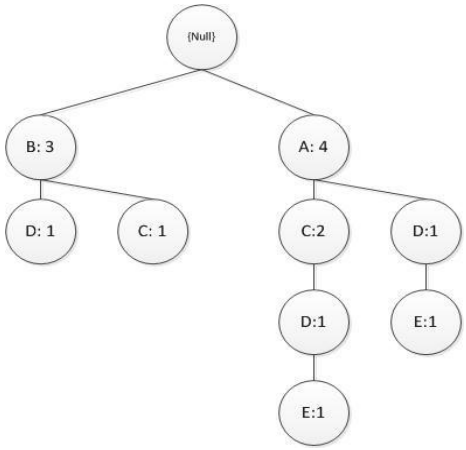
Selanjutnya pada gambar di bawah ini memberikan ilustrasi mengenai pembentukan *FP-Tree* setelah pembacaan TID 2



Gambar 2. 4 Hasil Pembentukan *FP-Tree* setelah pembacaan TID 2



Gambar 2.5 Hasil Pembentukan *FP-Tree* setelah pembacaan TID 5



Gambar 2.6 Hasil pembentukan *FP-Tree* setelah pembacaan TID 7

Diberikan 7 data transaksi dengan 5 jenis item seperti pada tabel di atas. Gambar 5-7 menunjukan proses terbentuknya *FP-Tree* setiap TID dibaca. Setiap simpul pada FP-Tree mengandung nama sebuah item dan *counter support* yang berfungsi untuk menghitung frekuensi kemunculan item tersebut dalam tiap lintasan transaksi.

Setelah *FP-Tree* terbentuk dari sekumpulan data transaksi maka, proses pencarian *frequent itemset* dengan menggunakan algoritma *FP-Growth* akan dilakukan yang dibagi menjadi tiga tahapan utama, yaitu:

a. Tahap pembangkitan *conditional pattern base*

   Merupakan subdata yang berisi *prefix path* (lintasan awal) dan *suffix pattern* (pola akhiran). Pembangkitan *conditional pattren base* didapatkan melalui *FP-Tree* yang telah dibangun sebelumnya.

b. Tahap pembangkitan *conditional FP-Tree*

   Pada tahap ini, *support count* dari setiap *item* pada setiap *conditional pattern base* dijumlahkan, lalu setiap *item* yang memiliki jumlah *support count* lebih besar atau sama dengan *minimum support count* akan dibangkitkan dengan *conditional FP-Tree.*

Tabel 2. 3 Hasil Perhitungan *Conditional Pattern Base* dan *Conditional FP-Tree*

Item	*Conditional Pattern Base*	*Conditional FP Tree*
E	{{A,C,D:1}, {A,D:1}}	{<A:2,D:2>}
D	{{B:1}, {A,C:1}, {A:1}}	{A:2}
C	{B:1}, {A:2}	{A:2}

c. Tahap pencarian *frequent itemset.*

   Apabila conditional *FP-Tree* merupakan lintasan tunggal(*single-path*), maka didapatkan *frequent itemset* dengan melakukan kombinasi *item* untuk setiap *conditional FP-Tree*. Jika bukan lintasan tunggal, maka dilakukan pembangkitan *FP-Growth* secara *rekursif* (proses memanggil dirinya sendiri) seperti contoh pada tabel 2.3 dibawah ini :

Tabel 2. 4 *Hasil Frequent Itemset*

Suffix	*Frequent Itemset*
E	E, A-E, D-E, A-D-E
D	D, A-D
C	C, A-C

Tabel 2.5 *Hasil* Generate Rule

Rule	*Support*	*Confidence* (%)
A → E	2	(2/4)*100 = 50
E → A	2	(2/2)*100 = 100
A → D	2	(2/4)*100 = 50
D → A	2	(2/3)*100 = 66.67
A → C	2	(2/4)*100 = 50
C → A	2	(2/3)*100 = 66.67
D → E	2	(2/3)*100 = 66.67
E → D	2	(2/2)*100 = 100
A- D →E	2	(2/2)*100 = 100
D- E →A	2	(2/2)*100 = 100

2.2.7 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program *berorientasi objek* serta aplikasinya. Berikut adalah beberapa model yang digunakan dalam perancangan *Data Mining* Pemaketan Produk di Minimarket Warga Tunggal untuk menggambarkan sistem dalam UML:

1. Diagram *Use Case*
2. Diagram *Aktivity*
3. Diagram *Class*
4. Diagram *Sequence*

2.2.7.1 Diagram *Use Case*

Diagram *use case* adalah model *fungsional* sebuah sistem yang menggunakan aktor dan *use case*. *Use case* adalah layanan *(services)* atau fungsi–fungsi yang disediakan oleh sistem untuk penggunanya.

Deskripsi Diagram *Use Case:*

1. Sebuah *use case* adalah dimana sistem digunakan untuk memenuhi satu atau lebih kebutuhan pemakai.
2. *Use case* merupakan awal yang sangat baik untuk setiap *fase* pengembangan berbasis objek, design testing, dan dokumentasi.
3. *Use case* menggambarkan kebutuhan sistem dari sudut pandang di luar sistem.
4. Use case menentukan nilai yang diberikan sistem kepada pemakainya.
5. *Use case* hanya menetapkan apa yang seharusnya dikerjakan oleh sistem, yaitu kebutuhan fungsional sistem.
6. *Use case* tidak untuk menentukan kebutuhan nonfungsional, misal: sasaran kerja, bahasa pemrograman.

2.2.7.2 Diagram *Class*

Diagram kelas adalah diagram UML yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan operasi. Tahapan dari diagram kelas adalah sebagai berikut:

1. Mengidentifikasi objek dan mendapatkan kelas-kelasnya.
2. Mengidentifikasi atribut kelas-kelas.
3. Mulai mengkonstruksikan kamus data.
4. Mengidentifikasi operasi pada kelas-kelas.
5. Mengidentifikasikan hubungan antar kelas dengan menggunakan *asosiasi*, *agregasi*, dan *inheritance* (pewarisan).

2.2.7.3 Diagram Aktifitas

Diagram aktivitas adalah representasi grafis dari seluruh tahapan alur kerja. Diagram ini mengandung aktivitas, pilihan tindakan, perulangan dan hasil dari aktivitas tersebut. Diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen suatu sistem.

2.2.7.4 Diagram Objek

Sequence diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message. Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/even untuk menghasilkan output tertentu. Sequence Diagram diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

2.2.8 MySQL

MySQL adalah *multiuser database* yang menggunakan bahasa *structured Query Language* (SQL). *MySQL* dalam operasi *client server* melibatkan *server daemon MySQL* di sisi *server* dan berbagai macam program serta *library* yang berjalan di sisi *client. MySQL* mampu menangani data yang cukup besar. Perusahaan yang mengembangkan *MySQL* yaitu TcX, mengaku bahwa *MySQL* yang mampu menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 *Gigabyte* data [2].

SQL adalah bahasa standar yang digunakan untuk mengakses *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai bahasa yang standar industri. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan menggunakan DBASE atau *Clipper* yang masih menggunakan perintah-perintah pemrograman.